11

Protection of Biometric Information

Tom Kevenaar

11.1 Introduction

The field of biometrics is concerned with recognizing individuals by means of unique physiological or behavioral characteristics. In practical systems, several biometric modalities are used, such as fingerprint, face, iris, hand geometry, and so forth. Recently, biometric systems are becoming increasingly popular because they potentially offer more secure solutions than other identification means such as PIN codes and security badges because a biometric is tightly linked to an individual. For the same reason, biometrics can prevent the use of several identities by a single individual. Finally, biometrics are also more convenient because, unlike passwords and PIN codes, they cannot be forgotten and are always at hand.

In this chapter we describe how biometrics can be combined with cryptographic techniques described in Part I of this book in order to, for example, derive cryptographic keys from biometric measurements or to protect the privacy of information stored in biometric systems.

In order to do so, this chapter is organised as follows. The remainder of this section gives an overview of a general biometric verification system. The privacy threats introduced by traditional biometric systems are presented in Section 11.2 and requirements for a private biometric system are given in Section 11.3. Section 11.4 gives a general architecture and Section 11.5 gives several implementations of an important building block in the architecture (the quantizer). Then, in Section 11.6, security and privacy consideration are discussed and the chapter concludes with a number of application examples that are based on biometric template protection techniques.

11.1.1 Overview of Biometric Systems

In this subsection we describe biometric systems and introduce terminology that is required in subsequent sections of this chapter.
A biometric system can be used for verification and identification of individuals. In verification, a person claims to have a certain identity and the biometric system performs a 1:1 comparison between the offered biometric and the biometric reference information that is linked to the claimed identity and stored in the biometric system. In identification, a 1:N comparison is performed between the offered biometric template and all available reference information stored in the biometric system to reveal the identity of an individual. Without loss of generality we will only consider verification systems as the 1:N comparison in an identification system is, in general, implemented as a sequence of 1:1 comparisons. Figure 11.1 gives a high-level overview of a biometric system. During enrollment, an acquisition device ACQ (e.g. a fingerprint sensor) measures a biometric. After processing the measurement data and extracting relevant features in the feature extraction (FE) block, a template representing the measurement is stored in the biometric system. During verification, an individual claims an identity, and a biometric measurement from this individual is obtained. This measurement is transformed into a template and compared (matched) with the template stored in the biometric system corresponding to the claimed identity and an “Accept” or “Reject” message is generated.

In the following subsection, the matching process (MATCH) will be explained in more detail.

11.1.2 Statistical Classification

In biometric applications, the measurements and the resulting templates are inherently noisy. Apart from the noise present in the measurement system, this is mainly due to varying interaction of the biometric with the acquisition device (sensor). For example, a different pressure on a finger when offering it to a fingerprint sensor will lead to distortion of the fingerprint image. Likewise, different lighting conditions in a face recognition systems will result in variations in the acquired measurements. Although a large proportion of the
variations can be eliminated in the feature extraction phase (FE), the generated biometric templates will always contain a certain amount of noise.

Consequently, comparison or matching of biometric templates must be treated as a statistical classification process that determines if a measured template is drawn from the probability distribution of the claimed identity (the genuine distribution) or from the distribution describing all other individuals (the impostor distribution or background distribution). To this purpose we assume that a biometric template can be represented as a feature vector \( f \in \mathbb{R}^l \) that is an observation of the stochastic variable \( F \). We have two classes \( \omega_I \) and \( \omega_G \) with probability density functions \( p(F|\omega_I) \) and \( p(F|\omega_G) \), respectively. Thus, given \( p(F|\omega_I) \) and \( p(F|\omega_G) \), the classifier (or matcher) must assign the observation \( f \) to \( \omega_I \) or \( \omega_G \) so as to minimize the probability of assigning \( f \) to the wrong class. In other words, a decision criterion or classification boundary must be determined against which \( f \) must be tested in order to assign \( f \) to \( \omega_I \) or \( \omega_G \).

A decision criterion based on a posteriori probabilities [115] chooses the class \( \omega_i, i \in \{I,G\} \) that is most probable for this given \( f \), or

\[
P(\omega_I|f) \geq P(\omega_G|f) \rightarrow f \in \{\omega_I, \omega_G\}.
\] (11.1)

If \( P(\omega_i) \) denotes the a priori probability that an event from class \( \omega_i \) occurs, then we can use Bayes’ rule

\[
P(\omega_i|f) = \frac{p(f|\omega_i)P(\omega_i)}{p(f)} \tag{11.2}
\]

to rewrite (11.1) as

\[
p(f|\omega_I)P(\omega_I) \geq p(f|\omega_G)P(\omega_G) \rightarrow f \in \{\omega_I, \omega_G\} \tag{11.3}
\]
or

\[
l(f) = \frac{p(f|\omega_G)}{p(f|\omega_I)} \geq \frac{P(\omega_I)}{P(\omega_G)} \rightarrow f \in \{\omega_G, \omega_I\}. \tag{11.4}
\]

The term \( l(f) \) is called the likelihood ratio and is the basic quantity in hypothesis testing. From (11.4) it is clear that the classification boundary is given by

\[
l(F) = \frac{p(F|\omega_G)}{p(F|\omega_I)} = \frac{P(\omega_I)}{P(\omega_G)}; \tag{11.5}
\]

so depending on which side of the boundary the observation \( f \) is located, it is assigned to \( \omega_I \) or \( \omega_G \). Eq. (11.4) is called the Bayes test for minimum error.

In general, for any classification boundary, there will be occurrences that the observation \( f \) is assigned to the wrong class, leading to a classification
error. In order to evaluate the performance of a given decision rule or classification boundary, the probability of error $\epsilon$ must be determined. Let us denote by $\Gamma_G$ and $\Gamma_I$ the regions where $p(f|\omega_G)/p(f|\omega_I) > P(\omega_I)/P(\omega_G)$ and $p(f|\omega_G)/p(f|\omega_I) < P(\omega_I)/P(\omega_G)$, respectively. Note that if $f \in \Gamma_I$, it will be assigned to class $\omega_I$ (likewise for $\Gamma_G$ and $\omega_G$). In general, we have

$$\epsilon = \Pr(\text{error}) = P(\omega_G)\Pr\{\text{error}|\omega_G\} + P(\omega_I)\Pr\{\text{error}|\omega_I\}. \quad (11.6)$$

The term $\Pr\{\text{error}|\omega_I\}$ is the probability that $f$ originated in class $\omega_I$ but is assigned to $\omega_G$ (likewise for $\Pr\{\text{error}|\omega_G\}$) so we have that the probability of a wrong classification of an observation $f$ is

$$\epsilon = P(\omega_G)\int_{\Gamma_I} p(F|\omega_G) \, dF + P(\omega_I)\int_{\Gamma_G} p(F|\omega_I) \, dF. \quad (11.7)$$

In biometrics, often the expressions False Accept Rate (FAR) and False Reject Rate (FRR) are used. The setting in which these entities are used is closely related to two-class classification described earlier. In our notation, $\omega_G$ is associated with “Accept” and $\omega_I$ with “Reject” and we have

$$\text{FAR} = \int_{\Gamma_G} p(F|\omega_I) \, dF \quad (11.8)$$

and

$$\text{FRR} = \int_{\Gamma_I} p(F|\omega_G) \, dF. \quad (11.9)$$

An example for two one-dimensional Gaussian distributions is given in Fig. 11.2. Where (11.8) and (11.9) give the FAR and FRR for a single individual, most commonly FAR and FRR are reported as an average over all available individuals.

In many practical situations including biometrics, the a priori probabilities $P(\omega_I)$ and $P(\omega_G)$ are not known and (11.4) cannot be used as a decision criterion. An alternative is to choose a decision rule based on the required FAR.
and FRR. One could, for example, minimize FRR over all possible decision rules under the constraint that $\text{FAR} \leq \alpha$, where $\alpha$ is a value chosen depending on the application. The solution of this constrained optimization problem is known as the Neyman-Pearson criterion [115] and defines the decision rule based on the likelihood ratio as

$$l(f) = \frac{p(f|\omega_G)}{p(f|\omega_I)} \gtrsim \eta \rightarrow f \in \left\{ \begin{array}{ll} \omega_G, \\
\omega_I \end{array} \right\},$$

(11.10)

where $\eta$ is chosen such that

$$\int_{\forall F | l(F) > \eta} p(F|\omega_I) \, dF = \alpha.$$  

(11.11)

By modifying the threshold $\eta$, a trade-off between FAR and FRR can be achieved. The quality of a classifier can be represented as a Receiver Operating Characteristic (ROC) curve that gives a trade-off between FAR and FRR (see Fig. 11.3). In order to represent the quality of such curve by a single number, usually the Equal Error Rate (EER) is chosen, defined as the point where the FAR equals the FRR.

Due to complexity reasons or the fact that $p(F|\omega_G)$ and $p(F|\omega_I)$ cannot be accurately estimated, one sometimes resorts to simpler decision rules such as Euclidean distance, defined as

$$|f - \mu|^2 \gtrsim \eta \rightarrow f \in \left\{ \begin{array}{ll} \omega_I, \\
\omega_G \end{array} \right\},$$

(11.12)

where $\mu$ is the mean of $p(F|\omega_G)$.  

---

**Fig. 11.3.** An illustration of a ROC curve (solid line).
In the following sections, biometric templates \( \mathbf{f} \) will be represented as binary strings \( \mathbf{z} \in \{0,1\}^* \) such that they can more easily be combined with the techniques in Part I. A common decision rule for these binary templates is based on the Hamming distance such that

\[
d_{HD}(\mathbf{z} - \mu_z) \geq \eta \rightarrow \mathbf{z} \in \left\{ \omega_I, \omega_G \right\},
\]

where \( \mu_z \) is the binary string representation of the biometric of an individual. Most biometric modalities are not naturally represented as binary strings and, therefore, in Section 11.5 some methods will be given to transform feature vectors into binary strings.

From the explanation given above it follows that biometric systems store biometric reference information in the form of biometric templates. In the following section we discuss privacy issues that arise when there is a widespread use of biometric systems.

### 11.2 Privacy Threats of Biometrics

A biometric template is a representation of a unique characteristics of an individual, and as such, it contains privacy-sensitive information. Especially when biometric systems store this information without any precaution in centralized databases or on unsecure devices, privacy problems will arise.

The first problem is that biometrics might contain information on the health condition of an individual [35,223]. Second, when an attacker obtains a biometric template, he might impersonate the rightful owner of the template by spoofing the biometric because it is well known that based on biometric templates, fake biometric identifiers can be produced that pass the identification test [186]. This will lead to identity theft. This problem becomes even more serious if we realize that biometric templates cannot be renewed or reissued (e.g., [244]) because people have only ten fingers, two eyes, and so forth. This stands in sharp contrast to situations where passwords and tokens are used for verification because they can easily be reissued. Third, when the biometric templates are stored without adequate protection in a database, they can be used to perform cross-matching between databases and track people’s behavior. A malicious employee of a bank can, for instance, find out that some biometric templates in his database also appear in the database of a night club. The underlying notion is that it is not possible to obtain an “alias” of a biometric template as is possible with, for example, names and addresses. Finally, in many countries, legislation obliges institutions to properly protect the stored personal information and to properly regulate who has access to which kind of information.

The above-mentioned privacy problems become less severe if we assume that a database owner (or a verifier) can be trusted, in which case the privacy
protection of biometric information hinges on employees correctly following procedures, but in practical situations, this is difficult to maintain. In the past years, much research has been carried out to protect the privacy of biometric information using technological means rather than procedural means. These technical methods are collectively referred to as biometric template protection techniques. An elaborate overview of such techniques is given in Chapter 10 as well as a possible approach known as Cancelable Biometrics. In this chapter we describe how biometrics can be combined with cryptographic techniques described in Part I of this book, but we start by giving an overview of the requirements for template protection.

11.3 Requirements for Template Protection

In this section we first consider two approaches that might be considered to achieve template protection. From the drawbacks of these approaches and the discussion in Section 11.2 we then will derive the security requirements for template protection.

11.3.1 Naive Approaches

One might think that encryption of biometric templates before storing them in a biometric system solves the problem. We show here that a straightforward application of encryption does not solve the privacy problem with respect to the verifier.

Assume that we use a symmetric key encryption scheme (the system works similarly for a public-key scheme). All sensors get a secret key $K$ that is equal to the secret key of the verifier. During enrollment, a biometric template $f$ of a person is obtained, $f$ is encrypted with the key $K$, and $E_K(f)$ is stored in the biometric system. During verification, the measurement of the same biometric results in the value $f'$ (close but not identical to $f$ due to noise). The verification system (e.g., the sensor) encrypts the value $f'$ with the key $K$ and sends $E_K(f')$ to the verifier. The verifier is faced with the problem of comparing $E_K(f)$ with $E_K(f')$. However, encryption functions have the property that $E_K(f)$ and $E_K(f')$ are very different even when $f$ and $f'$ are very close (but not equal). Hence, given only the values $E_K(f)$ and $E_K(f')$, the verifier cannot decide whether $f$ and $f'$ originate from the same person. This implies that the verifier must decrypt $E_K(f)$ and $E_K(f')$ to obtain $f$ and $f'$ and find out whether they are sufficiently similar. However, in that case, the verifier knows $f$ and, hence, the system does not provide privacy with respect to the verifier. Furthermore, in practical situations, the need for a cryptographic key infrastructure will severely inhibit a wide acceptance of this approach.

The problem of storing reference information also exists with password authentication. In order to protect passwords against the owner of the database and eavesdropping, the following measures are taken. During enrollment,
a cryptographic hash function $H$ is applied to a chosen password $pwd$ and the hash of the password $H(pwd)$ together with the username or identity $ID$ is stored in the (public) database for authentication. For example, in the UNIX system, this database can be found in the directory `/etc/passwd`. During authentication, the identity $ID$ and the password $pwd'$ are entered and $(ID, H(pwd'))$ is sent to the verifier. The verifier then compares $H(pwd')$ with $H(pwd)$, and when $H(pwd) = H(pwd')$, access is granted to the computer, otherwise access is denied. The security of this system follows from the fact that $H$ is a one-way function: Given $H(pwd)$, it is very hard to compute $pwd$. Hence, for the owner of the database as well as for the eavesdropper, it is infeasible to retrieve $pwd$ from $H(pwd)$.

In essence, one would like to mimic the password authentication scheme in the case of biometrics. The problem is, as explained in Section 11.1, that biometrics are inherently noisy and that $H$ is a one-way function. These functions are very good for security purposes but have no continuity properties. Applying the password authentication scheme implies that $H(f)$ is stored in the reference database. During authentication, the value $f'$ is obtained, which is typically close to $f$ when $f$ and $f'$ originate from the same person, but, in general, they are not equal due to noise. Therefore, due to the one-way property of $H$, even when $f$ and $f'$ are very close, $H(f)$ and $H(f')$ will be very different.

In the remainder of this chapter it will be explained how the various techniques explained in Part I can be used to protect biometric information. Before that, we give some security assumptions and requirements for template protection systems.

### 11.3.2 Security Assumptions and Requirements

The scenarios in the previous subsections illustrate that an encryption approach to template protection does not work because the verifier must be trusted. Hashing biometric templates is not feasible because biometric measurements are inherently noisy. In order to come up with a template protection system, the following security assumptions are made:

- Enrollment is performed at a Trusted Authority (TA). The TA enrolls all users by capturing their biometrics, performing additional processing and adding a protected form of the user data to a database.
- The storage is vulnerable to attacks both from the outside and from the inside (malicious verifier).
- During the authentication phase, an attacker is able to present artificial biometrics at the sensor.
- All capturing and processing during authentication is tamper-resistant; for example, no information about biometrics can be obtained from the sensor. The sensor is assumed to be trusted; it does not distribute measured information.
• The communication channel between the sensor and the authentication authority is public; that is, the line can be eavesdropped by an attacker.

In essence, this means that template protection methods protect the storage of biometric systems against attackers. The requirements for an architecture that does not suffer from the threats mentioned in Section 11.2 are given as follows:

• The information that is stored in the biometric system does not give sufficient information to make successful impersonation possible.
• The information in the biometric system provides the least possible information about the original biometrics; in particular, it reveals no sensitive information about the persons whose biometrics are stored.
• During matching, a verifier should not have access to (unprotected) biometric templates. This means that, at the verifier, a secure sensor is used that does not output unprotected biometric information.
• It is possible to generate several representations of a single biometric template.
• When a biometric measurement of the same person is contaminated with noise, verification (or identification) should still be successful if the noise is not too large.

Note that an approach that meets those requirements guarantees that the biometric cannot be compromised, it prevents cross-matching, and it can handle noisy biometric measurements. In the following section it is explained how template protection, in principal, can be achieved using some of the methods given in Part I.

11.4 An Architecture for Biometric Template Protection

Some of the methods in Part I derive cryptographic keys from noisy binary strings. In the key derivation process, side information $W$ is stored (or published), which makes it possible to retrieve the key from a noisy version of the original binary string. For binary strings with certain properties of their probability distribution (e.g., a minimum value for the min-entropy), bounds can be derived for the number of secure key bits that can be extracted and the amount of information leakage.

As explained in Section 11.1.2, most biometrics can be represented as real-valued feature vectors $f$ in a high-dimensional space $\mathbb{R}^l$. If it would be possible to represent these feature vectors $f$ accurately enough as binary strings, these methods could be used to protect biometric information stored in biometric systems.

A high-level architecture of a template protected biometric system is given in Fig. 11.4. As compared to Fig. 11.1, a quantizer $Q$, a key extractor $EXTR$, and a cryptographic protocol $PROT$ is added. The quantizer $Q$ transforms
feature vectors $f \in \mathbb{R}^l$ into binary strings $z \in \{0,1\}^*$, which can serve as an input for the extractor. These quantizers are not commonly used in traditional biometric systems nor are they part of the theory of (fuzzy) extractors, which assumes that binary strings with a certain probability distribution are given. Therefore, in Section 11.5 some quantizers will be discussed. In order to work properly, some quantizers generate side information $W^{(Q)}$ during enrollment, which is used during verification. This is indicated by a dotted line in Fig. 11.4. Given the binary representation of a biometric, the extractor EXTR can extract a key $K$. The extractor EXTR will generate exactly the same key $K$ if for two successive inputs $z$ and $z'$, the distance $d(z,z') < t$, where $d$ is some metric and $t$ is a user-defined threshold incorporated in EXTR. Therefore, in order to perform a biometric verification, the key $K$ generated during enrollment should be matched exactly with the key $K'$ generated during verification. Note that if the metric $d$ used in the extractor is the Hamming distance, matching the binary strings $z$ and $z'$ is in effect a Hamming distance classifier (see also (11.13)).

The fact that $K$ and $K'$ must be compared exactly makes it possible to use the large range of cryptographic authentication protocols (PROT in Fig. 11.4) for biometric authentication. Depending on the attack model, one could simply store the hash $h(K)$ and, during authentication, compare $h(K)$ and $h(K')$. An other possibility is to use zero-knowledge protocols. Choosing, for example, Schnorr’s zero-knowledge protocol implies that $g^{K \mod p}$ is stored in the biometric system, where $p$ is a prime and $g$ is a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$, and during verification, the sensor proves knowledge of $K$. The essence of privacy protection of biometric information is that the public
information required for the cryptographic authentication protocol (such as $h(K)$ or $g^K \mod p$) does not leak information about $K$. Thus, provided that $W^{(Q)}$ and $W$ do not leak information, the privacy of the biometric information is protected.

Most extractors EXTR are or can be changed into randomized functions. This means that from a single input $z$, it is possible to derive several different keys that also results in different side information stored in the biometric system. This means that several representations of a biometric can be generated. The randomization information is considered to be part of $W$.

Provided the side information for the quantizers $W^{(Q)}$ and the side information $W$ for the key derivation process only leak a limited amount of information about $f$ or $z$ and provided that the algorithms during verification are executed on a secure sensor, this architecture fulfills the requirements mentioned in Section 11.3.2. The issue of information leakage will be discussed in Section 11.6.

11.5 Quantization of Biometric Measurements

The architecture explained in Section 11.4 requires a quantizer that transforms feature vectors $f \in \mathbb{R}^l$ into binary strings $z \in \{0,1\}^*$ while allowing some “noise” or “fuzziness” in the input. More specifically, during enrollment, one or more biometric measurements are used to derive a binary string $z$ that is stored in the biometric system in a protected form. During verification, a binary string $z'$ is derived from a biometric measurement and compared with the string $z$. The general notion is that methods such as fuzzy extractors are insensitive to noise (e.g., derive the same cryptographic key) as long as the noise is smaller, in terms of the Hamming distance, than some preset threshold.

In the context of binary strings derived from biometric measurements, this means that the Hamming distance between $z$ and $z'$ should be small enough. Moreover, the binary templates should lead to proper classification results in terms of FAR, FRR, and EER (see Section 11.1.2). In this section we will discuss methods from the literature that transform feature vectors $f$ into binary strings.

11.5.1 Shielding Functions

One of the first methods to extract binary strings from continuous distributions was given in [182] and the method was inspired by a method called Quantization Index Modulation to embed a watermark in an audio or video stream [62]. It will be illustrated how a single bit is generated for the $t$-th feature $(f_i)_t$ in the feature vector $f_i$ of user $i$. The complete binary string $z_i$ is obtained by repeating the procedure for all entries in $f$, followed by concatenation of all the bits derived from the individual features.
Fig. 11.5. An illustration of the shielding function approach for a feature for which a “1” should be embedded.

Enrollment

For enrollment of user $i$, we assume that an estimate for the genuine distribution $p_{i,t}(F | \omega_G)$ (with mean $(\mu_{i})_t$) and the background distribution $p_{t}(F | \omega_I)$ is available as depicted in Fig. 11.5. In most practical situations, these estimates are obtained from a number of measurements of user $i$ and a number of measurements of other users, respectively. Note that the feature axis is uniformly quantized using a fixed quantization step $q$ and that every other interval is labeled using a “0” or a “1”. In order to embed a bit $(z^i)_t$, the value $W_{i,t}^{(Q)}$ is computed as

$$W_{i,t}^{(Q)} = \begin{cases} (2n + \frac{1}{2})q - (\mu_{i})_t & \text{if } (z^i)_t = 1 \\ (2n - \frac{1}{2})q - (\mu_{i})_t & \text{if } (z^i)_t = 0, \end{cases}$$

where $n \in \mathbb{Z}$ such that $-q < W_{i,t}^{(Q)} < q$. To finalize the enrollment, the binary string $z_i$ for user $i$ is defined as the concatenation of the bits $(z^i)_t$, $t = 1, \ldots, l$, and the values $W_{i,t}^{(Q)}$, $t = 1, \ldots, l$, are stored in the biometric system and used to extract $(z'_i)_t$ during verification.

Verification

During verification, a feature $(f'_i)_t$ is obtained and the extracted bit $(z'_i)_t$ is computed as

$$(z'_i)_t = \begin{cases} 1 & \text{if } 2nq \leq (f'_i)_t + W_{i,t}^{(Q)} < (2n + 1)q \\ 0 & \text{if } (2n - 1)q \leq (f'_i)_t + W_{i,t}^{(Q)} < 2nq, \end{cases}$$

for all $n \in \mathbb{Z}$. 
It can be seen that a bit error contributes to the FRR of the system. The bit error probability for the shielding function approach is

\[
FRR_{i,t} = \sum_{n=-\infty}^{\infty} \int (2^{n+\frac{3}{2}})^q p_{i,t}(F - (\mu_i)_t|\omega_G) \, dF.
\]  (11.16)

Likewise, the probability that an impostor generates the proper bit contributes to the FAR and equals \( \text{FAR}_{i,t} = 0.5 \).

11.5.2 Reliable Components

An alternative method also extracts a single bit per feature and was first published in [268]. It uses the estimate of the error probability of an extracted bit to select the most reliable bits to be incorporated in the binary string \( z \). As in Section 11.5.1, the bit extraction process for the \( t \)-th feature \( (f_i)_t \) of user \( i \) will be explained first which is followed by an explanation on how to select the most reliable bits.

Enrollment

Again, we assume that for enrollment of user \( i \), an estimate for the genuine distribution \( p_{i,t}(F|\omega_G) \) (with mean \((\mu_i)_t\) and standard deviation \(\sigma_{i,t}^2\)) and the background distribution \( p_t(F|\omega_I) \) (with mean \((\mu)_t\)) is available as depicted in Fig. 11.6. Given \((\mu_i)_t, \sigma_{i,t}^2\), and \((\mu)_t\), a (candidate) bit \((q_i)_t\) is extracted according to

\[
(q_i)_t = \begin{cases} 
0 & \text{if } (\mu_i)_t \leq (\mu)_t \\
1 & \text{if } (\mu_i)_t > (\mu)_t.
\end{cases}
\]  (11.17)

![Fig. 11.6. An illustration of extracting a bit from the \( t \)-th feature of user \( i \), including the reliability \( R_{i,t} \) of this bit.](image)
Next, the reliability of a bit \((q_i)_t\) is estimated as
\[
R_{i,t} = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{|(\mu_i)_t - (\mu)_t|}{\sqrt{2\sigma_{i,t}^2}} \right) \right),
\]
where \(\text{erf}\) is the error function.

Finally, the binary string \(z_i\) for user \(i\) is generated by choosing the most reliable components from the candidate bits \((q_i)_t\). The ordered set \(W^{(Q)}_i\) contains the indexes of the reliable bits in \(q_i\) and is stored in the biometric system and used to derive \(z'_i\) during verification.

**Verification**

During verification, a feature vector \(f'_i\) is measured, and for every entry \((f'_i)_t\), a candidate bit \((q'_i)_t\) is generated according to
\[
(q'_i)_t = \begin{cases} 
0 & \text{if } (f'_i)_t \leq (\mu)_t \\
1 & \text{if } (f'_i)_t > (\mu)_t.
\end{cases}
\]

Finally, using \(W^{(Q)}_i\), the appropriate bits are selected from the candidate bits \((q'_i)_t\) to form the string \(z'\).

Clearly, the estimated error probability equals \(\text{FRR}_{i,t} = 1 - R_{i,t}\) and the probability that an impostor generates the same bit equals \(\text{FAR}_{i,t} = 0.5\).

### 11.5.3 Multiple-Bit Extraction

The methods explained in the previous sections extract one bit per feature. In some cases,—for example, when the standard deviation \(\sigma_{i,t}\) of \(p_{i,t}(F|\omega_G)\) is much smaller than the standard deviation \(\sigma_i\) of \(p_i(F|\omega_I)\)—extracting more than one bit per feature might lead to better classification results (see also Section 11.5.4). In [309], the authors discussed an approach to extract more than one bit per feature and this method will be discussed in this subsection.

**Enrollment**

Like the methods discussed earlier, for enrollment of user \(i\), the current method assumes that an estimate for the genuine distribution \(p_{i,t}(F|\omega_G)\) (with mean \((\mu_i)_t\) and standard deviation \(\sigma_{i,t}\) and the background distribution \(p_{i}(F|\omega_I)\) is available, as depicted in Fig. 11.7. Based on these distributions, a parameterized acceptance interval \(A^{(0)}_{i,t}\) is defined as
\[
A^{(0)}_{i,t} = [(\mu_i)_t - k_{i,t}\sigma_{i,t}, (\mu_i)_t + k_{i,t}\sigma_{i,t}]
\]
as well as some other intervals $A_{i,t}^{(1)}, A_{i,t}^{(2)}, \ldots$. If there are $a_{i,t}$ intervals, every interval can be encoded using a bitstring $(z_t)_t$ containing $\lceil \log_2(a_{i,t}) \rceil$ bits. For every feature, the interval boundaries and the binary codes per interval are stored in the biometric system. However, before storing the intervals, an optimal value for $k_{i,t}$ must be determined as explained below.

Area I in Fig. 11.7 indicates the false accept rate ($\text{FAR}_{i,t}$) and areas II and III together depict the false reject rate ($\text{FRR}_{i,t}$). Thus, the parameter $k_{i,t}$ allows for a trade-off between $\text{FAR}_{i,t}$ and $\text{FRR}_{i,t}$, as it defines the width of the acceptance interval.

If we assume that all $l$ features in the vector $f$ are independent, the total FAR and FRR are given as

$$\text{FAR}_i = \prod_{t=1}^{l} \text{FAR}_{i,t}$$

and

$$(1 - \text{FRR}_i) = \prod_{t=1}^{l} (1 - \text{FRR}_{i,t}).$$

In order to find optimal values for $\text{FAR}_i$ and $\text{FRR}_i$ for user $i$, optimal values for the parameters $k_{i,t}, t = 1, \ldots, l$, can be obtained by solving the following optimization problem:

$$\max_{k_{i,t}, t=1 \ldots l} (1 - \text{FRR}_i)$$

subject to $\text{FAR}_i \leq \alpha, 0 \leq \alpha \leq 1.$

Using (11.21) and (11.22) and setting the Lagrange multiplier $\lambda \geq 0$, this optimization problem can be formulated as
By choosing a value for $\lambda$ and maximizing the expressions $\log(1 - \text{FRR}_{i,t}) - \lambda \log(\text{FAR}_{i,t})$ individually, the optimal values for $k_{i,t}$, $i = 1, \ldots, l$, are obtained along with the optimal values for $(1 - \text{FRR}_{i,t})$ and $\text{FAR}_{i,t}$. Using (11.21) and (11.22), one thus obtained a single point on the optimal ROC curve of $(1 - \text{FRR}_i)$ versus $\text{FAR}_i$. By varying $\lambda$ and repeating this procedure, different points on this optimal ROC curve can be computed from which a point of operation can be chosen.

Finally, the binary representation $z_i$ of the vector $f_i$ is defined as the concatenation of the binary codes $(z_{i,t})_t$, $t = 1, \ldots, l$, coding for the acceptance intervals $A_{i,t}^{(0)}$, $A_{i,t}^{(1)}$, $A_{i,t}^{(2)}$, ..., $A_{i,t}^{(l)}$.

**Verification**

During verification, a feature vector $f_i'$ is measured and every entry $(f_{i,t}')$ is assigned to one of the intervals $A_{i,t}^{(0)}$, $A_{i,t}^{(1)}$, $A_{i,t}^{(2)}$, ... The corresponding binary code is looked up in the biometric system. All $l$ binary codewords are concatenated to yield the string $(z_{i,t}')_t$, the binary representation of $f_i'$.

### 11.5.4 Discussion on Classification Results

In the previous subsections, methods were presented to transform an enrollment feature vector $f$ and a verification feature vector $f'$ into binary strings $z$ and $z'$, respectively. These binary strings can be used in the general architecture for biometric template protection depicted in Fig. 11.4, where a biometric verification is successful if $K = K'$. If we assume that the extractor EXTR will generate the same key $K$ if, for two successive inputs $z$ and $z'$, the distance $d(z, z') < t$, where $d$ is the Hamming distance and $t$ is a user-defined threshold incorporated in EXTR, then matching the binary strings $z$ and $z'$, is in effect a Hamming distance classifier (see also (11.13)).

From the viewpoint of a biometric system it is important to consider the classification results (e.g., in terms of EER) for the system in Fig. 11.4. At first sight one might expect that the methods in Sections 11.5.1–11.5.3, which quantize every feature to one or a few bits, will significantly deteriorate the classification results. However, simulations using several real databases (see Table 11.1 and [309]), suggest that quantization does not have a major impact on classification quality. Nevertheless, further research has to be done to
Table 11.1. Some classification results for real fingerprint and face databases

<table>
<thead>
<tr>
<th>Feature</th>
<th>EER</th>
<th>Feature Vector</th>
<th>Binary String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprints Database 1 (see [268])</td>
<td>1.4%</td>
<td>4.0%</td>
<td></td>
</tr>
<tr>
<td>Fingerprints Database 2 (see [268])</td>
<td>1.6%</td>
<td>3.5%</td>
<td></td>
</tr>
<tr>
<td>Face Database 1 (see [170])</td>
<td>1.5%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>Face Database 2 (see [170])</td>
<td>0.25%</td>
<td>0.25%</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 11.8. Distributions for the assumption of a Gaussian channel.

enhance the quality of the binary strings and thus the classification performance of the template protected biometric system.

There are two intuitions supporting the relatively good classification results of the binary strings. The first one is concerned with the information content of the individual features. Let us assume that the \( t \)th feature of individual \( i \) is denoted by \((\mu_i)_t\) and that the distribution of the values \((\mu_i)_t\) over the whole population is given by \(p_t(M_i)\) with variance \(\sigma^2_{M,t}\). Further assume that for every user \( i \), the genuine (noise) distribution is Gaussian according to \(N((\mu_i)_t, \sigma^2_t)\) such that every user for this feature has the same variance but a different mean (see Fig. 11.8). Under these assumptions, we have a Gaussian channel and the maximum amount of information that can be transmitted over this channel is limited by [66]

\[
\frac{1}{2} \log \left(1 + \frac{\sigma^2_{M,t}}{\sigma^2_t}\right). \quad (11.28)
\]

As an illustration, Fig. 11.9 gives the results for the FVC2000 database [185] containing 8 gray-scale images of 110 different fingers captured at 500 dpi, resulting in an image size of \(256 \times 364\) pixels. Feature vectors \( \mathbf{f} \) were derived containing 1536 entries using methods described in [11–13]. It can be seen that the channel capacity varies between 0.2 and 1.9 bits, with an average of 0.79 bits. Although this clearly is not a scientific proof and the assumption for a Gaussian genuine distribution does not always hold, it gives an intuition why
Fig. 11.9. Simulation results for the channel capacity for each of 1536 features derived from fingerprint images.

quantizing a feature to a single or a few bits does not necessarily deteriorate the classification results.

The second intuition is related to a phenomenon known as overtraining, which might occur when a model is used that has too many parameters compared to the number of training data available. If the redundancy in the parameters is not properly dealt with, the model will not behave well for observations that were not in the training set. As explained in Section 11.1.2, in traditional systems, often an estimate for the high-dimensional probability distributions \( p(F|\omega_I) \) and \( p(F|\omega_G) \) must be obtained using, in general, a limited amount of training data. This might result in inaccurate estimates of the distributions and, thus, in an overtrained and non-optimal classifier. The quantization methods described in Sections 11.5.1–11.5.3 assume that all of the features are independent such that only scalar distributions have to be estimated. Although the assumption of independent features is, in general, not correct, practical simulations show that this assumption can be less severe than assuming that an accurate estimate of distributions is available.

11.6 Security and Privacy Considerations

In discussing the privacy and security of biometric systems, privacy refers to the required effort to obtain the biometric information of an individual. This is related to the amount of information in the biometric itself (e.g., expressed
as the mutual information $I$ between $z$ and $z'$) and the information leakage (or entropy loss) due to the data stored in the biometric system (e.g., $W$ and $W^{(Q)}$ in Fig. 11.4). On the other hand, security refers to the required effort to be accepted by a biometric system as a certain individual without having access to the biometric of this individual.

The discussions on the privacy and security of practical template-protected biometric systems are mostly complicated by the fact that up until now, little is known about the statistical properties of biometric templates and, in our case, about the binary templates $z$ derived from feature vectors $f$ in terms of, for example, Shannon entropy, min-entropy, dependency of the bits in $z$, and so forth. In several publications, security and privacy bounds are given assuming some (statistical) properties of the binary strings $z$. If the strings $z$ are of finite length but independent and identically distributed (i.i.d.), it can be shown [270] that the information leakage equals $nh(p)$, where $h$ is the Shannon entropy function and $p$ is the probability that two corresponding bits in $z$ and $z'$ are different. Other authors [90] assume that the probability distribution of the strings $z$ has a certain min-entropy and give constructions (secure sketches) for which the information leakage (entropy loss) of the data stored in the biometric system is bounded by the redundancy in the Error-Correcting Code used in the secure sketch. Under the same assumption, they also derive bounds regarding the number of secure key bits that can be extracted from a biometric measurement.

Unfortunately, in practical biometric systems the assumptions do not hold or it is not known if they hold. Furthermore, known bounds on entropy loss and maximum possible key length are not always tight and this is especially problematic in situations in which sources have limited information content, as is the case with biometrics. This is in contrast to other practical applications such as Physical Unclonable Functions (PUFs) described in Chapters 12 and 13, which can be designed to have a high information content. Therefore, for example, bounds on entropy loss do not necessarily lead to practically relevant results about the entropy that remained in the biometric after observing its secure sketch.

Therefore, first, it is important to further study the properties of the binary strings $z$ that are derived from biometric measurements. Initial work in [46] estimates min-entropy values for binary strings derived from continuous distributions—for example, obtained by methods discussed in Sections 11.5.1–11.5.3 thus linking real-life continuous biometric distributions to methods like fuzzy extractors. The authors further aim to relate the min-entropy of the strings $z$ to the FAR, thus formalizing the intuition that the min-entropy of an extracted key (in bits) cannot be larger than $-\log_2(FAR)$.

This last point motivates research into improving the FAR (i.e., the classification results) of biometric systems, especially related to the strings $z$, at acceptable FRR values. Currently, most biometric modalities (except irises) have EERs in the order of 1% (see, e.g., [187]), which is too high to derive PIN codes, passwords, or keys of sufficient length. Still, it is probably not to be
expected that the information content of most biometric modalities is more
than, say, 100 bits. This means that tighter bounds are required on entropy
loss and maximum achievable key lengths in order to get meaningful results.

11.7 Application Examples of Template-Protected
Biometric Systems

This section discusses three practical application examples [169] based on
the architecture given in Fig. 11.4. In principle, the techniques for template
protection only store side information $W$ and but do not store biometric infor-
mation nor the key that is derived from it. This gives rise to architectures that
are different from the architectures when traditional, non-protected biometric
systems are used.

11.7.1 A Server Access Token

Introduction and Problem Definition

Corporate computing is becoming increasingly mobile as can be seen, for
example, from an increase in the notebook to desktop shipment ratio of
PCs [167]. This means that an increasing number of employees connect to
their corporate network from a remote location. Traditionally, access to these
corporate networks has been protected using a so-called server access token.
These tokens, often implemented as a small keyring device, typically contain a
secret key, an accurate time reference, and a small LCD display. Based on the
key and the time reference, some (cryptographic) function generates random
access codes that appear in the display of the token. The corporate network
also knows the secret key, and using an accurate time reference, it can verify
if a proper token is used. In combination with a PIN code assigned to the
owner of the token, access will be allowed to the corporate network.

In order to increase the convenience of using such a token, biometrics could
be used rather than PIN codes. A preferred implementation would be to equip
the token with a biometric sensor such that any PC could be used to log onto
the corporate network.

A straightforward approach is to store biometric templates in the token,
and if the offered biometric is close enough to the reference information, the
token will send a random access code derived from the key and the time
reference to the display. This approach has the drawback that biometric infor-
mation and the secret key are stored in the token and both can, in principle,
be retrieved by reverse engineering. Another possibility is to attack the point
in the device where the decision on the similarity of the stored and measured
biometric is made.

The following subsection gives a solution for these problems using Fig. 11.4.
Deriving the Secret Key from the Biometric

In this subsection, we propose an architecture for securing the server access token using biometrics. The solution is such that neither biometric reference information nor the secret key is stored on the device. In order to obtain a personalized server access token, an employee goes to the corporate IT department, where the following steps are performed (enrollment):

- A biometric (e.g., fingerprint) is measured several times, resulting in a enrollment string $z$.
- The extractor EXTR internally chooses a random value and generates $W$, which is stored on the token.
- The extractor EXTR generates a key $K$ that is stored in the database of corporate IT. The time reference of the token is synchronized with the reference of corporate IT.

When the employee wants to log onto the corporate network, he puts the proper finger on a sensor on the token and a binary string $z'$ is obtained (see Fig. 11.10). The token computes $K'$ using $W$ and and $z'$ and combines this with a time reference to form a random access code $ac_t$ that is shown on the display of the token.

At the moment of log-on, corporate IT combines the stored value $K$ with a local time reference giving an access code $ac_c$. If $z' \approx z$, we have $ac_t = ac_c$ and log-on is allowed. The proposed architecture enhances the security of the token because the secret key is not stored on the token but derived from a biometric, thus thwarting a physical attack on the token.

11.7.2 Three-Way Check for Biometric ePassport

Introduction and Problem Definition

Nowadays, most biometric applications store biometric reference information on a personal smartcard. An example is the Privium system [227] used for
automatic border passage at Schiphol Airport; reference information of an iris scan is stored on a personal Privium card.

However, many applications would benefit from storing biometric reference information on a central server or in a centralized database. One of the reasons is that this might lead to more secure applications because not all reference information is placed in the hands of possibly malicious individuals. The International Civil Aviation Organization (ICAO) [149] recently proposed an optional three-way check for the new biometric ePassport, for which a live biometric measurement is not only checked against the information on the passport but also against reference information stored in a database. In many countries, legislation allows storing biometric information in centralized databases provided that (complicated) procedures are put in place regulating access to the stored information. However, public opinion and privacy interest groups still can delay or prevent the use of databases.

In the following subsection we propose an architecture for a three-way check around the biometric ePassport, for which the reference information stored in a database contains no information on the biometric. Although the architecture will be explained for the ePassport, many other applications could benefit from this architecture. One could imagine an aircraft boarding system for which the boarding card contains secure biometric information and for which a three-way check is performed against the passenger list. Another example is a soccer stadium entrance system for which supporters are checked against a list of hooligans.

Architecture for a Three-Way Check

The architecture for the three-way check is given in Fig. 11.11, where Kiosk represents the location where a passport is checked. In order to explain the architecture, we assume that when the passport is issued, secure biometric information of the form \( (h(K_c), W) \) is stored in the passport and reference information of the form \( h(K) \) is stored in a database. A three-way check then proceeds as follows:

- The Kiosk reads \( (h(K_c), W) \) from the passport and sends \( W \) to the Sensor.
- The Sensor performs a biometric measurement resulting in a binary string \( z' \) and combines this with \( W \) to generate a key \( K_s \). The hash \( h(K_s) \) is sent to the Kiosk.
- If \( h(K_s) \neq h(K_c) \), authentication fails, otherwise the individual is considered to be the owner of the passport.
- Next, the Kiosk verifies if \( h(K_s) \) is in the database. Depending on the response, the Kiosk allows or denies the individual’s access.

Note that \( K \) is chosen independently from the biometric measurement \( z \) and, thus, \( h(K) \) can reveal no information on \( z \). Clearly, it is possible to add additional information, such as name and address, to a stored value \( h(K_i) \), but for the biometric part of the system, this is not required.
11.7.3 A Secure Password Vault

Introduction and Problem Definition

Nowadays, people have to remember a large number of passwords and PIN codes. In addition to PIN codes for bank accounts, the average computer user has to remember several different passwords for access to e-mail, internet accounts, web services, and so forth. Remembering all of these passwords is inconvenient, especially when systems also require passwords to be changed frequently. Moreover, strong passwords are random sequences in which all allowed characters have an equal probability of being used. This, however, makes them inherently hard to remember for humans.

In order to make life easier, a user often chooses passwords that are easy to remember or writes passwords on a piece of paper that is kept close to the log-in terminal. Both methods are insecure: Easy-to-remember passwords can be guessed without much effort and using a piece of paper allows a malicious individual to just read the passwords from the paper such that all of the user’s passwords are compromised at once.

We propose to solve these problems by introducing a Secure Password Vault (SPV) that uses biometrics and the architecture of Fig. 11.4.
Architecture for a Secure Password Vault

The SPV is a small device that is easily carried by the user and consists of the following components: a biometric sensor (e.g., a fingerprint sensor), a small display for showing a password or PIN to the user, some memory for storing helper data, and a processing unit to process the biometric data that are read from the sensor. Optionally, the SPV contains input means (e.g., a keyboard) allowing the user to choose a password and/or a random generator to generate random passwords. An example architecture of an SPV with keyboard and random generator is given in Fig. 11.12.

The basic idea behind the SPV is that it displays its user’s password or PIN code whenever the user presents his biometric to the sensor. This is achieved by storing the appropriate $W$ in the memory of the SPV during the initialization phase, which contains the following steps:

- The user presents his biometric to the sensor (resulting in a binary string $z$) and enters the desired password $K$ via the keyboard (alternatively, the SPV could use its Random Number Generator to generate a (strong) PIN or password $K$);
- Given $z$ and $K$, the SPV generates $W$ and stores it in memory (in many practical implementations of Fig. 11.4, it is possible to derive $W$ given $K$ and $z$).

Clearly, the SPV could also connect to an external device that enrolls the user, computes the appropriate helper data $W$, and stores it in the SPV’s memory.

During operation, the user presents his biometric to the sensor and the SPV obtains a string $z'$. Subsequently, the SPV combines $z'$ and $W$, resulting in a reconstructed password or PIN code $K'$, and, finally, $K'$ is displayed on the display of the SPV such that the user can read the result.

Note that the actual password $K$ is not stored in the memory of the SPV but is reconstructed from stored information $W$ and an offered biometric, $z'$. 

Fig. 11.12. Architecture for Secure Password Vault.
For any offered biometric, the device will produce a password, but only for the genuine biometric will it generate the correct password.

Obviously, by storing different values $W$ in the SPV, multiple passwords can be retrieved for different applications and the user could select the required password by choosing the application. Finally, as a practical implementation, the SPV could be built into another personal device such as a mobile phone or Personal Digital Assistant (PDA).

### 11.8 Conclusions

In this chapter we described how biometrics can be combined with some of the cryptographic techniques described in Part I of this book. A general architecture was given to derive PIN codes, passwords, or cryptographic keys from (noisy) biometric measurements. An important block in the architecture, the quantizer, was treated in some more detail, and, based on the proposed architecture, some practical application examples were given. A discussion on the privacy and security properties of the general architecture showed that the limited knowledge on the statistical properties of biometrics and the fact that bounds are not always tight makes it hard to give realistic quantitative results on privacy and security properties of practical systems. However, with the current classification performance for most biometric modalities, it is not to be expected that sufficiently long cryptographic keys can be derived. This motivates further research into the (statistical) properties of biometrics, tighter bounds, and better classifications results.